



Apache HTTP Server Version 1.3

Is this the version you want? For more recent versions, check our [documentation index](#).

Using Apache With Microsoft Windows

This document explains how to install, configure and run Apache 1.3 under Microsoft Windows. Most of this document assumes that you are installing Windows from a binary distribution. If you want to compile Apache yourself (possibly to help with development, or to track down bugs), see [Compiling Apache for Microsoft Windows](#).

If you find any bugs, please document them on our [bug reporting page](#). Contributions are welcomed, please submit your code or suggestions to the [bug report page](#), or join the [new-`httpd`](#) mailing list.

The [bug reporting page](#) and [new-`httpd`](#) mailing list are *not* provided to answer questions about configuration or running Apache. Before you submit a bug report or request, first consult this document, the [Frequently Asked Questions](#) page and the other relevant documentation topics. If you still have a question or problem, post it to the [comp.infosystems.www.servers.ms-windows](#) newsgroup, where many Apache users and several contributions are more than willing to answer new and obscure questions about using Apache on Windows.

[groups.google.com](#)'s newsgroup archive offers easy browsing of previous questions. Searching the newsgroup archives, you will usually find your question was already asked and answered by other users!

Warning: Apache on NT has not yet been optimized for performance. Apache still performs best, and is most reliable on Unix platforms. Over time NT performance has improved, and great progress is being made in the upcoming version 2.0 of Apache for the Windows platforms. Folks doing comparative reviews of webserver performance are still asked to compare against Apache on a Unix platform such as Solaris, FreeBSD, or Linux.

- [Requirements](#)
- [Downloading Apache for Windows](#)
- [Installing Apache for Windows \(binary install\)](#)
- [Running Apache for Windows](#)
- [Testing Apache for Windows](#)
- [Configuring Apache for Windows](#)
- [Running Apache in a Console Window](#)
- [Controlling Apache in a Console Window](#)
- [Running Apache for Windows as a Service](#)
- [Controlling Apache as a Service](#)
- [Compiling Apache for Microsoft Windows](#)

Requirements

Apache 1.3 is designed to run on Windows NT 4.0 and Windows 2000. The binary installer will only work with the x86 family of processors, such as Intel's. Apache may also run on Windows 95 and 98, but these have not been tested. In all cases TCP/IP networking must be installed.

If running on NT 4.0, installing Service Pack 3 or 6 is recommended, as Service Pack 4 created known issues with TCP/IP and WinSock integrity that were resolved in Service Pack 5 and later.

Note: "Winsock2" is required for Apache 1.3.7 and later.

If running on Windows 95, the "Winsock2" upgrade must be installed before Apache will run.

"Winsock2" for Windows 95 is available [here](#) or via [here](#). Be warned that the Dialup Networking 1.2 (MS DUN) updates include a Winsock2 that is entirely insufficient, and the Winsock2 update must be reinstalled after installing Windows 95 dialup networking. Windows 98, NT (Service Pack 3 or later) and 2000 users need to take no special action, those versions provide Winsock2 as distributed.

Downloading Apache for Windows

Information on the latest version of Apache can be found on the Apache web server at <http://httpd.apache.org/>. This will list the current release, any more recent alpha or beta-test releases, together with details of mirror web and anonymous FTP sites.

You should download the binary build of Apache for Windows named as `apache_1_3_#-win32-src.msi` if you are interested in the source code, or simply `apache_1_3_#-win32-no_src.msi` if you don't plan to do anything with the source code and appreciate a faster download. Each of these files contains the complete Apache runtime. You must have the Microsoft Installer version 1.10 installed on your PC before you can install the Apache runtime distributions. Windows 2000 and Windows ME are both delivered with the Microsoft Installer support, others will need to download it. For more information, visit the main download page at <http://httpd.apache.org/download.cgi>. Instructions on locating the Microsoft Installer, as well as the binary distributions of Apache, are found at [the win32 download directory on the mirrors](#).

The source code is available in the `-src.msi` distribution, or from the [distribution directory](#) as a `.zip` file. If you plan on compiling Apache yourself, there is no need to install either `.msi` package. The `.zip` file contains only source code, with MS-DOS line endings (that is cr/lf line endings, instead of the single lf used for Unix files distributed in `.tar.gz` or `.tar.Z` archives.)

While the source is also available as a `.tar.gz` `.tar.Z` archive, these contain unix lf line endings that cause grief for Windows users. To use those archives, you must convert at least the `.mak` and `.dsp` files to have DOS line endings before MSVC can understand them. Please stick with the `.zip` file to spare yourself the headache.

Note: prior to 1.3.17 Apache was distributed as an InstallShield 2.0 `.exe` file. With an increasing number of users unable to run the InstallShield package [on Windows ME or Windows 2000] the binaries were repackaged into the readily available Microsoft Installer `.msi` format.

Installing Apache for Windows

Run the Apache .msi file you downloaded above. This will prompt you for:

- whether or not you want to run Apache for all users (installing Apache as a Service), or if you want it installed to run in a console window when you choose the Start Apache shortcut.
- your Server name, Domain name and administrative email account.
- the directory to install Apache into (the default is C:\Program Files\Apache Group\Apache although you can change this to any other directory you wish)
- the installation type. The "Complete" option installs everything, including the source code if you downloaded the -src.msi package. Choose the "Custom" install if you choose not to install the documentation, or the source code from that package.

During the installation, Apache will configure the files in the `conf` directory for your chosen installation directory. However if any of the files in this directory already exist they will **not** be overwritten. Instead the new copy of the corresponding file will be left with the extension `.default.conf`. So, for example, if `conf\httpd.conf` already exists it will not be altered, but the version which would have been installed will be left in `conf\httpd.default.conf`. After the installation has finished you should manually check to see what is new in the `.default.conf` file, and if necessary update your existing configuration files.

Also, if you already have a file called `htdocs\index.html` then it will not be overwritten (no `index.html.default` file will be installed either). This should mean it is safe to install Apache over an existing installation (but you will have to stop the existing server running before doing the installation, then start the new one after the installation is finished).

After installing Apache, you should edit the configuration files in the `conf` directory as required. These files will be configured during the install ready for Apache to be run from the directory where it was installed, with the documents served from the subdirectory `htdocs`. There are lots of other options which should be set before you start really using Apache. However to get started quickly the files should work as installed.

If you eventually uninstall Apache, your configuration and log files will not be removed. You will need to delete the installation directory tree ("C:\Program Files\Apache Group" by default) yourself if you do not care to keep your configuration and other web files. Since the `httpd.conf` file is your accumulated effort in using Apache, you need to take the effort to remove it. The same happens for all other files you may have created, as well as any log files Apache created.

Running Apache for Windows

There are two ways you can run Apache:

- As a "service". This is the best option if you want Apache to automatically start when your machine boots, and to keep Apache running when you log-off.
- From a console window. Closing this console window will terminate the Apache server.

Complete the steps below before you attempt to start Apache as a Windows "service"!

To run Apache from a console window, select the "Start Apache as console app" option from the Start menu (in Apache 1.3.4 and earlier, this option was called "Apache Server"). This will open a console window and start Apache running inside it. The window will remain active until you stop Apache. To

stop Apache running, either press select the "Shutdown Apache console app" icon option from the Start menu (this is not available in Apache 1.3.4 or earlier), or see [Controlling Apache in a Console Window](#) for commands to control Apache in a console window.

In Apache 1.3.13 and above it is now quite safe to press Ctrl+C or Ctrl+Break to stop the Apache in the console window. And on Windows NT/2000 with version 1.3.13, Apache will stop if you select 'Close' from the system menu (clicking the icon on the top-left corner of the console window) or click the close (X) button on the top-right corner. The Close menu item and close (X) button also work on Windows 95/98 as of Apache version 1.3.15. But do *not* try any of these approaches on earlier versions of the Apache server, since Apache would not clean up.

Testing Apache for Windows

If you have trouble starting Apache please use the following steps to isolate the problem. This applies if you started Apache using the "Start Apache as a console app" shortcut from the Start menu and the Apache console window closes immediately (or unexpectedly) or if you have trouble starting Apache as a service.

Run the "Command Prompt" from the Start Menu - Programs list. Change to the folder to which you installed Apache, type the command apache, and read the error message. Then review the error.log file for configuration mistakes. If you accepted the defaults when you installed Apache, the commands would be:

```
c:  
cd "\program files\apache group\apache"  
apache  
Wait for Apache to exit, or press Ctrl+C  
more <logs\error.log
```

After looking at the error.log you will probably have a good chance of working out what went wrong and be able to fix the problem and try again. If you are unable to work it out then please follow the [guidelines for assistance](#) at the top of this document or in the [FAQ](#). Many users discover that the nature of the httpd.conf file is easier to manage and audit than page after page of configuration dialog boxes.

After starting Apache running (either in a console window or as a service) it will be listening to port 80 (unless you changed the Port, Listen or BindAddress directives in the configuration files). To connect to the server and access the default page, launch a browser and enter this URL:

`http://localhost/`

This should respond with a welcome page, and a link to the Apache manual. If nothing happens or you get an error, look in the `error.log` file in the `logs` directory. If your host isn't connected to the net, you may have to use this URL:

`http://127.0.0.1/`

Once your basic installation is working, you should configure it properly by editing the files in the `conf` directory.

Because Apache *CANNOT* share the same port with another TCP/IP application, you may need to stop or uninstall certain services first. These include (but are not limited to) other web servers, and firewall

products such as BlackIce. If you can only start Apache with these services disabled, reconfigure either Apache or the other product so that they do not listen on the same TCP/IP ports. You may find the Windows "netstat -an" command useful in finding out what ports are in use.

Configuring Apache for Windows

Apache is configured by files in the `conf` directory. These are the same as files used to configure the Unix version, but there are a few different directives for Apache on Windows. See the [Apache documentation](#) for all the available directives.

Begin configuring the Apache server by reviewing `httpd.conf` and its directives. Although the files `access.conf` and `srm.conf` both exist, these are old files which are no longer used by most administrators, and you will find no directives there.

`httpd.conf` contains a great deal of documentation itself, followed by the default configuration directives recommended when starting with the Apache server. Begin by reading these comments to understand the configuration file, and make small changes, starting Apache in a console window with each change. If you make a mistake, it will be easier to back up to configuration that last worked. You will have a better idea of which change caused the server to fail.

The main differences in Apache for Windows are:

- Because Apache for Windows is multithreaded, it does not use a separate process for each request, as Apache does with Unix. Instead there are usually only two Apache processes running: a parent process, and a child which handles the requests. Within the child each request is handled by a separate thread. So, "process"-management directives are different:
 - `MaxRequestsPerChild` - Like the Unix directive, this controls how many requests a process will serve before exiting. However, unlike Unix, a process serves all the requests at once, not just one, so if this is set, it is recommended that a very high number is used. The recommended default, `MaxRequestsPerChild 0`, does not cause the process to ever exit.
 - `ThreadsPerChild` - This directive is new, and tells the server how many threads it should use. This is the maximum number of connections the server can handle at once; be sure and set this number high enough for your site if you get a lot of hits. The recommended default is `ThreadsPerChild 50`.
- The directives that accept filenames as arguments now must use Windows filenames instead of Unix ones. However, because Apache uses Unix-style names internally, you must use forward slashes, not backslashes. Drive letters can be used; if omitted, the drive with the Apache executable will be assumed.
- Apache for Windows has the ability to load modules at runtime, without recompiling the server. If Apache is compiled normally, it will install a number of optional modules in the `modules` directory. To activate these, or other modules, the new `LoadModule` directive must be used. For example, to activate the status module, use the following (in addition to the status-activating directives in `httpd.conf` - see the [mod_status docs](#) for more details.):

```
LoadModule status_module modules/mod_status.so
```

Information on [creating loadable modules](#) is also available. Note that some 3rd party modules may be distributed in the old style names, `ApacheModuleFoo.dll`. Always set the `LoadModule` command as directed by the 3rd party module's own documentation.

- Apache for Windows version 1.3 series is implemented in synchronous calls. This poses an enormous problem for CGI authors, who won't see unbuffered results sent immediately to the browser. This is not the behavior described for CGI in Apache, but it is a side-effect of the Windows port. Apache 2.0 is making progress to implement the expected asynchronous behavior, and we hope to discover that the NT/2000 implementation allows CGI's to behave as documented.
- Apache can also load ISAPI Extensions (*i.e.*, Internet Server Applications), such as those used by Microsoft's IIS, and other Windows servers. More information is available. Note that Apache *CANNOT* load ISAPI Filters.

- When running CGI scripts, the method Apache uses to find the interpreter for the script is configurable using the ScriptInterpreterSource directive.
- Since it is often difficult to manage files with names like `.htaccess` under windows, you may find it useful to change the name of this configuration file using the AccessFilename directive.

Running Apache in a Console Window

The Start menu icons and the NT Service manager can provide a simple interface for administering Apache. But in some cases it is easier to work from the command line.

When working with Apache it is important to know how it will find the configuration files. You can specify a configuration file on the command line in two ways:

- `-f` specifies a path to a particular configuration file:

```
apache -f "c:\my server\conf\my.conf"  
apache -f test\test.conf
```

- `-n` specifies the configuration file of an installed Apache service (Apache 1.3.7 and later):

```
apache -n "service name"
```

In these cases, the proper `ServerRoot` should be set in the configuration file.

If you don't specify a configuration file name with `-f` or `-n`, Apache will use the file name compiled into the server, usually "conf/httpd.conf". Invoking Apache with the `-V` switch will display this value labeled as `SERVER_CONFIG_FILE`. Apache will then determine its `ServerRoot` by trying the following, in this order:

- A `ServerRoot` directive via a `-C` switch.
- The `-d` switch on the command line.
- The current working directory
- A registry entry, created if you did a binary install.
- The server root compiled into the server.

The server root compiled into the server is usually "/apache". invoking apache with the `-V` switch will display this value labeled as `HTTPD_ROOT`.

When invoked from the start menu, Apache is usually passed no arguments, so using the registry entry is the preferred technique for console Apache.

During a binary installation, a registry key will have been installed, for example:

```
HKEY_LOCAL_MACHINE\Software\Apache Group\Apache\1.3.13\ServerRoot
```

This key is compiled into the server and can enable you to test new versions without affecting the current version. Of course you must take care not to install the new version on top of the old version in the file system.

If you did not do a binary install then Apache will in some scenarios complain about the missing registry key. This warning can be ignored if it otherwise was able to find its configuration files.

The value of this key is the "ServerRoot" directory, containing the `conf` directory. When Apache starts it will read the `httpd.conf` file from this directory. If this file contains a `ServerRoot` directive which is different from the directory obtained from the registry key above, Apache will forget the registry key and use the directory from the configuration file. If you copy the Apache directory or configuration files to a new location it is vital that you update the `ServerRoot` directory in the `httpd.conf` file to the new location.

To run Apache from the command line as a console application, use the following command:

```
apache
```

Apache will execute, and will remain running until it is stopped by pressing control-C.

Controlling Apache in a Console Window

You can tell a running Apache to stop by opening another console window and running:

```
apache -k shutdown
```

Note: This option is only available with Apache 1.3.3 and later.

For earlier versions, you must use Control-C in the Apache console window to shut down the server.

From version 1.3.3 through 1.3.12, this should be used instead of pressing Control-C in a running Apache console window, because it allowed Apache to end any current transactions and cleanup gracefully.

As of version 1.3.13 pressing Control-C in the running window will cleanup Apache quite gracefully, and you may use `-k stop` as an alias for `-k shutdown`. Earlier versions do not understand `-k stop`.

You can also tell Apache to restart. This makes it re-read the configuration files. Any transactions in progress are allowed to complete without interruption. To restart Apache, run:

```
apache -k restart
```

Note: This option is only available with Apache 1.3.3 and later. For earlier versions, you need to use Control-C in the Apache console window to shut down the server, and then restart the server with the Apache command.

Another *very useful* feature is the configuration files test option. To test the Apache configuration files, run:

```
apache -t
```

This is especially useful following alterations to the configuration files while Apache is still running. You can make the changes, confirm that the syntax is good by issuing the "apache -t" command, then restart Apache with "apache -k restart". Apache will re-read the configuration files, allowing any transactions in progress to complete without interruption. Any new request will then be served using the new configuration.

Note: for people familiar with the Unix version of Apache, these commands provide a Windows equivalent to `kill -TERM pid` and `kill -USR1 pid`. The command line option used, `-k`, was chosen as a reminder of the "kill" command used on Unix.

Apache HTTP Server

